Jet calibration tool for 2011 data/MC

parton level jet

particle level jet
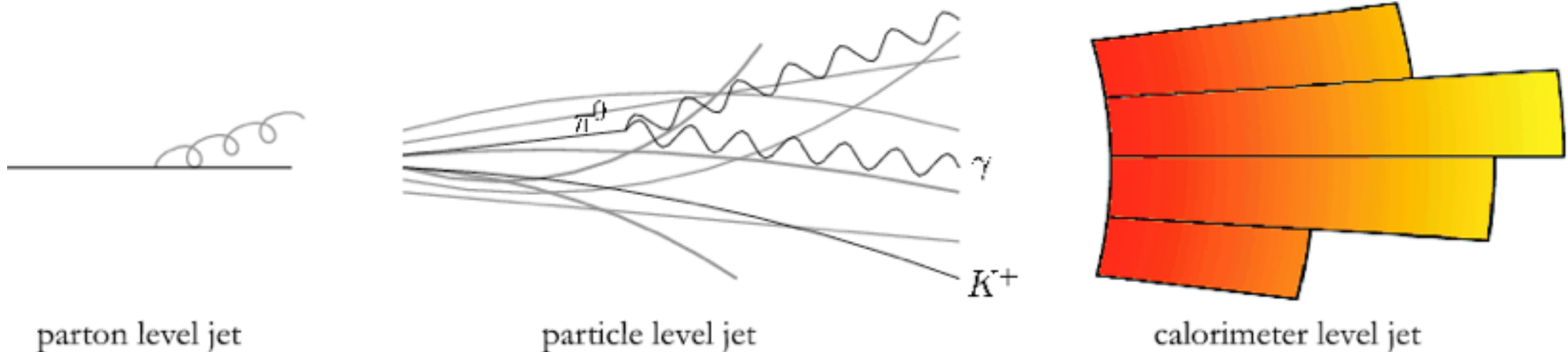
calorimeter level jet

Dag Gillberg

Carleton University

2011-12-10

ApplyJetCalibration

- Tool to apply the jet calibration on D3DP analysis

- Used for testing new jet calibrations and (possibly) as temporary solution before data is reprocessed with final calibration applied by default

- Applies both **absolute JES** (EtaJES), **offset** correction and **residual in-situ derived corrections** (not implemented yet)

- Can be checked out by:

  ```
  svn co $SVNGRP/CombPerf/JetETMiss/JetCalibrationTools/
  ApplyJetCalibration/tags/ApplyJetCalibration-00-00-01
  ```

- RootCore compatible

How it works - in user macro

jet algo   Calibration settings   Is this data? (false=MC)

```cpp
#include "ApplyJetCalibration/ApplyJetCalibration.h"

...

JetCalibrationTool myJES("AntiKt4TopoEM","Rel17_JES.config",true);

// obtain a calibrated jet
TLorentzVector jet = myJES.ApplyOffsetEtaJES(Eem,eta_det,eta,phi,m,
                                             mu,NPV);
```
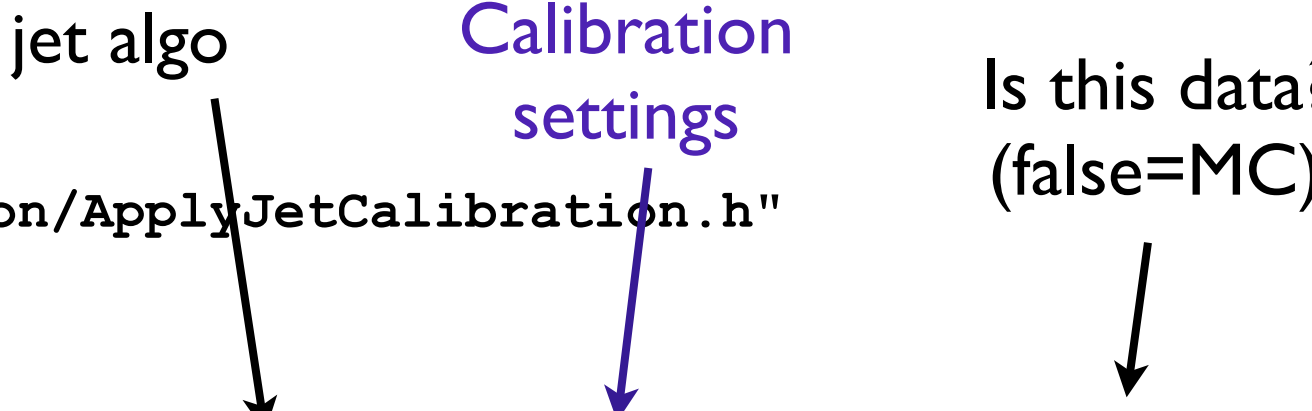
All energies are in MeV!

```cpp
// can also get part corrections only
double JES = myJES.GetJES(Eem,eta_det);

// offset correction
double offsetET = myJES.GetOffset(mu,NPV);

// offset correction, set muRef=0 and npvRef = 1
double offsetET = myJES.GetOffset(mu,NPV,0,1);
```

Example code for EM+JES

For use in simple D3PD analysis.

```cpp
TString JES_config_file="ApplyJetCalibration/CalibrationConfigs/Rel17_JES.config";
JetCalibrationTool *myJES = new JetCalibrationTool("AntiKt4TopoEM",JES_config_file);

for (Long64_t jentry=0; jentry<nentries;jentry++) {
    Long64_t ientry = LoadTree(jentry);
    if (ientry < 0) break;
    fChain->GetEntry(jentry);

    int Njets=jet_AntiKt4TopoEM_E->size();
    for ( int jeti=0; jeti<Njets; jeti++ ) {
        double Eraw     = jet_AntiKt4TopoEM_emscale_E->at(jeti);
        double eta_det  = jet_AntiKt4TopoEM_emscale_eta->at(jeti);
        double eta      = jet_AntiKt4TopoEM_EtaOrigin->at(jeti);
        double phi      = jet_AntiKt4TopoEM_PhiOrigin->at(jeti);
        double m        = jet_AntiKt4TopoEM_MOrigin->at(jeti);

        double mu=averageIntPerXing; int NPV=0;
        for ( unsigned tracki=0; tracki<vxp_nTracks->size(); tracki++)
            if ( vxp_nTracks->at(tracki) >= 2 ) NPV++;

        TLorentzVector jet = myJES->ApplyOffsetEtaJES(Eraw,eta_det,eta,phi,m,mu,NPV);

        ...
```

Example code for LC+JES

```cpp
TString JES_config_file="ApplyJetCalibration/CalibrationConfigs/Rel17_JES.config";
JetCalibrationTool *myJES_akt4lc = new JetCalibrationTool("AntiKt4LCTopo",JES_config_file);

for (Long64_t jentry=0; jentry<nentries;jentry++) {
    Long64_t ientry = LoadTree(jentry);
    if (ientry < 0) break;
    fChain->GetEntry(jentry);

    int Njets=jet_AntiKt4LCTopo_E->size();
    for ( int jeti=0; jeti<Njets; jeti++ ) {
        double Eraw    = jet_AntiKt4LCTopo_constscale_E->at(jeti);
        double eta_det = jet_AntiKt4LCTopo_emscale_eta->at(jeti);
        double eta     = jet_AntiKt4LCTopo_EtaOrigin->at(jeti);
        double phi     = jet_AntiKt4LCTopo_PhiOrigin->at(jeti);
        double m       = jet_AntiKt4LCTopo_MOrigin->at(jeti);

        double mu=averageIntPerXing; int NPV=0;
        for ( unsigned tracki=0; tracki<vxp_nTracks->size(); tracki++)
            if ( vxp_nTracks->at(tracki) >= 2 ) NPV++;

        TLorentzVector jet = myJES_akt4lc->ApplyOffsetEtaJES(Eraw,eta_det,eta,phi,m,mu,NPV);

        ...
```

The JES settings file

```
#
#  Settings for Rel 17 Jet calibration
#
###########################


# ---------------
# 1. Absolute JES
```

Two values for absolute JES:

```
  # The file with the absolute JES factors
AbsoluteJES.CalibFile:        CalibrationFactors/AbsoluteJES_Rel17.0.config
AbsoluteJES.Description:      JES for release 17 data derived from MC11_valid


# ---------------
# 2. Pile-up correction
```

A bit more for the offset correction:

```
  # What offset correction to apply
OffsetCorrection.Name:        OffsetMC11a

  # These corrections should correspond to the
  # conditions of the absolute JES calibration
OffsetCorrection.DefaultMuRef:  5.4
OffsetCorrection.DefaultNPVRef: 4.9

  # additional config files to include
Includes:            CalibrationFactors/MCBasedOffset.config


# ---------------
# 3. Residual in-situ correction - applied to data only!
```

```
#########
# EM+JES calibration factors for
# jet energy correction
#
JES.AntiKt4TopoEM_Bin0:          7.2851e-02   1
JES.AntiKt4TopoEM_Bin1:          2.6149e-01   9
JES.AntiKt4TopoEM_Bin2:          8.3079e-02   1
JES.AntiKt4TopoEM_Bin3:          1.8813e-01   1
JES.AntiKt4TopoEM_Bin4:          9.2332e-02   1
JES.AntiKt4TopoEM_Bin5:         -1.0183e-02   1
....
```

```
###########################
#
#  MC10b offset correction derived by Carlos Sand
#  for more details see:
#     https://twiki.cern.ch/twiki/bin/viewauth/Atl
#

OffsetMC10b.Description:  MC-based jet pile-up co

OffsetMC10b.AbsEtaBins:          0 0.3 0.8 1.2 2.1

OffsetMC10b.Slope.AntiKt4TopoEM:     0.297376 0.
OffsetMC10b.Intercept.AntiKt4TopoEM:  0.0262287 0

OffsetMC10b.Slope.AntiKt4LCTopo:      0.406266 0.
OffsetMC10b.Intercept.AntiKt4LCTopo:  0.0241725 0

OffsetMC10b.Slope.AntiKt6TopoEM:      0.554206 0.
OffsetMC10b.Intercept.AntiKt6TopoEM:  0.113253 0.

OffsetMC10b.Slope.AntiKt6LCTopo:      0.841648 0.
OffsetMC10b.Intercept.AntiKt6LCTopo:  0.157548 0.
```

Available calibrations

With links to performance plots

1. Rel 16.6 calibration with MC10b pile-up correction

   http://toshi.web.cern.ch/toshi/jetcalib/r2299/

2. Rel 17.0 calibration with latest MC11a pile-up correction

   http://toshi.web.cern.ch/toshi/jetcalib/r2529/

3. Rel 17.0 calibration after first subtracting the offset with latest MC11a pile-up correction

   http://toshi.web.cern.ch/toshi/jetcalib/r2731_offset_1.0_0.0/

All these calibrations were derived by Toshi Sumida

Default Rel 17 calibration

Run draw_each_JES.sh in the macros folder to make all this, and many more plots

JES for release 17 data derived from MC11_valid

Anti-k_t $R = 0.4$, EM+JES

Jet response at EM scale

- E = 30 GeV (green filled circle)
- E = 60 GeV (black open circle)
- E = 110 GeV (orange filled square)
- E = 400 GeV (blue open square)
- E = 2000 GeV (red filled triangle)

Jet η_{det}